Emergent information systems – the role of adaptive agents

Nathan T. Clapham  
School of Computer Science  
Australian Defence Force Academy  
Canberra ACT 2600 AUSTRALIA  
nclapham@cs.adfa.edu.au

David G. Green & Michael Kirley  
School of Environmental and Information Sciences  
Charles Sturt University  
Albury NSW 2640 AUSTRALIA  
{dgreen, mkirley}@csu.edu.au

Abstract

One of the greatest challenges facing the information age is the discovery of relevant information. Current approaches to the problem fall short in several respects. We propose a scalable, self-organising model for online distributed data warehouses. The prime feature of the model is a population of adaptive agents, one per web site. These agents produce reports by distributing elements of a query to appropriate resources or to other agents in the population, showing a form of emergent computation. Queries are couched in a specialised scripting language, based on XML. Agents adapt by converting new queries into generalised functions and by recording new resources.

Introduction

Everyone knows the scene from Star Trek. Confronted by the unknown the captain says: “Computer, tell me about...” and the computer instantly responds with a concise, lucid report giving exactly the necessary information. This looks so easy in science fiction. But in practice, query systems are a long way from rivalling such capabilities. The hurdles to be overcome are considerable.

We argue that it is possible to make practical headway by considering systems with restricted domains. Most importantly we argue that information systems need to employ agents that can adapt to the needs of their users. In this paper, we outline a model that uses adaptive information agents to solve some of the problems involved. We conclude by discussing some of the issues and implications that flow from the model.

Keywords: Information agents, information discovery, adaptive agents, emergent computation.

Resource discovery and data warehousing

The Internet has revealed the extent of the information age. With an estimated 800 million documents [16], the rapid growth of the Internet has placed a new complexion on the problem of compiling and organising information. This problem has now become one of the greatest hurdles for the effective use of the information now available.

Using current search methods, locating information on a relevant topic can be difficult. As opposed to organisational data warehouses the Internet is too dynamic and chaotic [9], both in terms of its structure and usage. In an attempt to organise information, automated robots such as Excite1 and Google2 construct some of the most popular indexes on the World Wide Web (WWW). Manual schemes also exist, such as Virtual Libraries (VL) and Information Networks [11].

Each of the search approaches listed above have in common a focus upon indexing information. This focus brings to light the issue that what people really want is information – not lists of web sites. Search engines, for instance, typically return lists containing thousands of items for each query. However, this response falls short in two crucial respects. First, the ratio of false “hits” is high. A simple query about the ‘flu virus, for example, would return thousands of links to items about computer viruses. Secondly, users themselves have to locate and extract the information they need from the returned documents. Manual indexes solve the problem of relevance, but cannot keep pace with the growing abundance of information.

Typically, a user wants a report that consists of elements drawn from several different distributed sources. For example, if a government officer wants to compile a report about (say) eucalyptus trees in Tasmania, then he or she might need taxonomic information from the national herbarium, photos from the Botanic Gardens, distribution maps from Environment Australia, and discussions of conservation from the state parks service. It is the distributed information (ie the final report) the user is interested in, not the list of resources.

Over the past decade or more many organisations have been working towards the goal of drawing existing information together in “data warehouses”. Examples cover a wide range of endeavours, including commercial records of marketing and sales, government documents, environmental management (eg [10]), and scientific data such as genomic information [2,5]. The advent of the Internet has accelerated the trend towards data warehousing; it has made possible the use of distributed data sources in which the information is

1 http://www.excite.com/
2 http://www.google.com/
stored and accessed seamlessly from a number of different sites. This distributed approach has an advantage that the organisations creating the data maintain control of it. It also provides an ability to handle far larger bodies of data than any one organisation could effectively manage by itself.

Protocols and standards

Protocols and standards are an integral component in the task of unifying information systems; however, XML, HTTP, Z39.50, and CORBA only partly address the challenges [13]. Nevertheless, several recent developments in XML standards are encouraging. These developments include specifications for XML Protocol, XML Query, and XML Path. XML Protocol provides a framework in which two or more sites may communicate using XML as an encapsulation language. This system provides a general and flexible messaging system. XML Query defines a standard data model for XML documents and a set of query operators. XML Path provides a language for addressing parts of XML documents. These standards are working towards a general and powerful model for sharing data among distributed sources.

The technical difficulties of defining protocols and standards are slowly being overcome. However, there is a need for standards to converge into a unified model. As different groups come together, working on the continual development of general standards, such as XML, CORBA, and metadata, such a unified body of standards may emerge. This unified model should define a framework of how distributed sites interact and share information, providing a strong foundation for the development of emergent information systems.

Emergent information systems

The Internet makes feasible collaborative projects on a scale not seen before. One of the first practical results was the appearance of information networks, in which a community of sites cooperate to provide a common information resource (eg [11]). However, of greater interest here is the creation of active data warehouses in which a single query may involve searching for and collating material from many different sites. To achieve this goal a practical framework is required. Software agents are a suitable framework, which have been shown to be an effective way of encapsulating the results of complex tasks into a single interface.

A key application of agent-based systems is the ability to produce or model emergent computation [7] in distributed environments. Emergent computation deals with the question of how local operations, preformed by individual agents, can produce some form of global result. The Internet is really a form of emergent computation, where local machines follow simple computational rules to produce a globally coordinated system. In large and complex environments, emergent computation is necessary as there is typically no possibility of centralised control. This is why centralised search indexes fail to be adequately created and maintained. Thus, we find a need to move away from centralised approaches to organising information to a collaborative emergent system. Such a system would use the information organisation and adaptation capabilities of individuals to contribute to the global system.

Another problem addressed by a multi-agent system that arises in such a distributed environment is the problem of separating the processing from the data resources. It can be highly inefficient to draw data together from many different sources and only then start to process the results. The alternative is to perform part of the processing at the source and only transmit the results, de-centralising the process of organising information. Such an arrangement of the above considerations calls for a new model, which we now describe.

An agent-based model

The distributed data warehouse (DDW) model described here is built around distributed query agents (DQA). These agents operate and adapt collaboratively. They provide the necessary facilities to separate the processing from the data resources. Also, the agent architecture is scalable, leaving scope for the larger system to grow. Agents are used here as a paradigm to develop collaborative interactions that produce a globally intelligent and adaptive information system.

In the basic DDW model (Fig. 1), the front end consists of a user interacting with an agent via typical WWW methods. This may be interactive, in real-time, or possibly batch style, allowing the agent more time to process the request while the user continues on. When handling a query an agent may either: (a) process the request itself, if it contains the required functions; or (b) send the request to another agent. Upon receipt of all requests, the agent compiles the results into a report, which is passed on to the user.

![Figure 1: Overview of the distributed data warehouse model.](image_url)

**Figure 1: Overview of the distributed data warehouse model.**
Some important characteristics of the basic model are:

- DQA’s represent data warehouses, each with a limited domain of interest, that is, agents are specialised in the information they organise; and,
- collaboration between agents provides the ability to learn of new data sources and to process queries extending beyond domain limitations.

These two characteristics allow a macro system to arise; seemingly providing information from possibly several data warehouses. This emergent system builds upon the existing design of the WWW by hiding the discovery of data resources from the end-user.

**The need for adaptive agents**

The other design issue involves the learning and adaptation of agents. Adaptation is essential because it is virtually impossible to design a system that adequately anticipates the broad range of user needs. Any system that can adapt has the potential for substantial improvements in efficiency and cost-effectiveness. In our model the system adapts automatically through the actions of individual agents, and through their interactions with one another. A key feature in the process of adaptation is the inclusion of a sleep and wake cycle for each agent. This feature exploits an analogy with the sleep and wake functions of a human. When humans sleep the brain sifts through the collected information to make sense of it and to store it as learned memories.

Likewise in our model an agent is ‘awake’ whenever it works on a query (whether triggered directly by a user, or indirectly through another agent). The details of each query are recorded in “short-term memory”. When not active, the agent reverts to a ‘sleeping’ or ‘dreaming’ mode, in which it processes the contents of its short-term memory.

The ‘dreaming’ processing consists of two complementary actions. One is to compare the elements of each query script against its current repertoire of functions. An agent’s functions are the procedures for acquiring information from a data source. Elements of the query that are identified as new are extracted to create new functions. This process consists chiefly of converting constants to variables. For instance if we could paraphrase a particular query as

"Produce a report about Eucalyptus regnans in Tasmania"

might generalise to

"Produce a report about <species> in <place>".

Notice that in this example we have replaced the constant Eucalyptus regnans with the variable <species>. To do this presupposes that the given name can be parsed as a plant species. Likewise it is essential to identify Tasmania as a place. The ability to do this is crucial since cruder generalisations, such as <object> instead of <species>, could allow the query to be applied to totally inappropriate data. Making these kinds of generalisations is difficult in a global context, but much simpler in the highly restricted domain of a data warehouse dealing with a limited range of themes. However, we make the process transparent by using an XML markup in which the constants are tagged with labels that indicate the appropriate variables.

The second adaptive process that agents carry out in sleeping mode is to link resources to queries. Thus, if a query links an existing function to a previously unknown WWW site or agent, then the details are added to a list of sources.

The model presented above outlines a scalable system. It deals with distributed data within restricted domains. Here lies the key. To implement practical systems that display the ability to adapt and learn, it is necessary to limit the domain. However, the scalable nature, through each agent’s ability to learn and collaborate, provides the foundation for useful intelligent and adaptive information system to emerge. See [6] for a detailed description of the distributed data warehouse model.

**Agent connectivity and emergence**

Maintaining macroscopic structural integrity is a key issue in any information system development. There is a need to decentralise information organisation yet maintain global coordination. An important question to be addressed is to determine the set of local rules, which govern local agent behaviour, which will lead to ‘useful’ emergent properties on a global scale. We call this type of system, where local interactions produce emergent properties, self-organising. This self-organisation, existing within the overall system, is crucial to unifying information systems, because, as we have seen it is virtually impossible to maintain centralised control.

The level of communication links between agents in the DDW model is an important variable of the emergent outcome and is the subject of continuing research. It is still an open question to investigate the various relationships of connectivity levels between agents and the nature of their local rules, along with the ability of individual adaptation. In the case of the DDW model, this correlates to number of connections each agent maintains and the methods of interaction between these connections.

There is some relevant research that highlights several implications relating to the connectivity structure of the DDW model. For example, Albert et. al. [1] shows that many real-world networks, with certain connectivity distributions, exhibit scale-free properties. That is, the relative size of the network (ie the average number of links required to traverse from agent $a$ to $b$) only slightly increases as the number of network components grow. Also, results in Stocker et. al. [19] suggest that it is possible for information networks to be self-organising providing connectivity between agents is sufficiently high, yet, without the need for each agent to be directly connected to every other agent. The implications for the distributed data warehouse model described here are numerous, for example:
♦ Agent's do not need to maintain a computationally expensive number of connections;
♦ the number of agents can grow without drastically affecting efficiency;
♦ information propagation through the network will be efficient; and,
♦ all nodes can be kept up-to-date.

The DDW model implements simple local interactions focusing on collaborating operations between agents. Each individual agent in the model distributes elements of a query that it cannot itself fulfill, and learns by recording new resources and generalising functions. These simple local interactions lead to an emergent adaptive information system, showing elements of both traditional rule-based artificial intelligence and fields such as artificial life that deal with emergence.

The implementation
To test the concepts of the DDW model described above, we developed a demonstration within a highly restricted domain. Here we describe the implementation and provide an example. The demonstration uses six agents to fulfill a query of the distribution of *Eucalyptus regnans* in Tasmania. The final product is a report containing a picture and description of the tree, a species distribution map, and a list of some relevant resources; these items are obtained from several data sources.

Each DQA was implemented using Java and resided behind a WWW server Common Gateway Interface (CGI). DQA's communicate through the CGI to fulfill a particular query. Queries are expressed in an XML markup (eg Fig. 3 below). We call this XML markup the Report Generation Language (RGL). The RGL expresses the information a report contains along with its structure.

The following is an RGL example of a map object within a report. Notice that within the map object a query tag is specified. This query tag details how the map object is to be retrieved. In this case the map object is retrieved from the source http://life.csu.edu.au/cgi-bin/specDistDQA.cgi (here the source refers to another agent; a species distribution agent), with the attributes of 'GENUS', 'SPECIES', and 'LOCATION'. The VAR tags refer to variables, which are defined using a TERM tag within the script (eg in Fig. 3).

```
<OBJECT TYPE="map">
  <QUERY SOURCE="http://life.csu.edu.au/cgi-bin/specDistDQA.cgi" TYPE="map">
    <ATTRIB ID="GENUS"><VAR ID="1"/></ATTRIB>
    <ATTRIB ID="SPECIES"><VAR ID="2"/></ATTRIB>
    <ATTRIB ID="LOCATION"><VAR ID="3"/></ATTRIB>
  </QUERY>
</OBJECT>
```

Perhaps the most important aspect of the model is the ability of the agents to learn and adapt. The RGL provides a simple mechanism for the adaptation and learning processes. Each agent maintains a knowledge-base of functions to retrieve information. Queries, such as the one given in the previous example, are mapped to a generalised function in the agents ‘sleeping’ cycle. Thus, the query above is mapped to the following function, which is maintained within an agent’s knowledge-base.

```
<FUNCTION TYPE="map" THEME="plant">
    <ATTRIB ID="GENUS"><VAR THEME="plant/genus"/></ATTRIB>
    <ATTRIB ID="SPECIES"><VAR THEME="plant/genus/species"/></ATTRIB>
    <ATTRIB ID="LOCATION"><VAR THEME="geographic/country/state"/></ATTRIB>
  </QUERY>
</FUNCTION>
```

Of particular importance is the THEME attribute of the VAR tag. This attribute indicates a generalised variable and was drawn from its initial definition. Now this function may be applied to a large range of queries fitting the generalised variables. For example, rather than information on *Eucalyptus regnans* in Tasmania, we may want information regarding *Eucalyptus melliodora* in New South Wales – which this generalised function would be able to perform. Thus, once agents have learnt functions from their received queries, they are able to produce an increasingly large range of reports without having to specify how to retrieve the information.

In the demonstration of a report on the species distribution of *Eucalyptus regnans* in Tasmania, six agents were used each contributing a specific role (eg Fig. 4). Firstly, an interface agent was constructed. This interface agent interacts with the user via a simple HTML form, as depicted in Fig. 2.

![Figure 2: A form generated by the interface agent.](image)

Upon completion of the form, the interface agent sends the query in Fig. 3 to a distributed query agent (call it plantDQA). Notice that this query does not specify how to retrieve the specified information, only what to retrieve. In this example the plantDQA has previously learnt from queries and contains, within its knowledge-base, the necessary functions to obtain the information the query expresses.
Figure 3: An example of the Report Generation Language.

Once the DQA has received the query it then proceeds to contact the necessary agents to acquire the desired information. When the agent plantDQA has gathered information from all other sources, the results are returned to the originating interface agent. The interface agent marks-up the results to HTML for display in a common WWW browser. Fig. 5 illustrates the final presentation.

As the final report illustrates, there are four separate components to the report, each drawn from a separate agent.

The communication between agents is shown in Fig. 4:
- imageDQA is an agent managing a data source of tree images;
- descDQA manages a data source of tree descriptions;
- specDistDQA manages species distribution data; and,
- resourceDQA manages a bookmark-type data source.

Discussion

The success of the Internet, the WWW and related information systems rests on identifying a simple and scalable model of how different sites communicate and share resources. With this in mind, what we have presented here is an attempt to develop a model of this kind for an intelligent and adaptive information system. These developments are only the first steps towards implementing this model. Many problems remain to be solved, and this account was intended only to demonstrate the feasibility of such a system.

One of the major suggestions in this paper was that it is a necessity to develop emergent information systems due to the complexity of the environment. There is a large body of research dealing in general with emergent computation that provides some theoretical understanding of the process. For instance, Langton [15] has studied the computational capabilities of cellular automata (CA) and suggests that computationally capable CAs exist within a certain 'critical' region of the CA rule table. This has implications for developing the rule-base of individual agents. Several other authors have also worked on models of emergent computation using cellular automata [3,817]. This work attempts to discover high-performance rules (ie those that produce emergent computation) and explain why these rules work. The outcome of these studies illustrate that the use of simple local interactions can produce complex and collaborative computational behaviours.

The restriction of the model to data warehouses, each with a restricted domain, is crucial in the first instance. A tightly constrained context conveys numerous advantages, including the assurance of standardised resources and the ability to restrict the domain and range of queries. Secondly, this restriction provides a framework for collaboration and emergent behaviour, where the combination of local interactions leads to a larger adaptive system. To some extent we have tied together studies of emergent phenomena and traditional artificial intelligence. Nevertheless, many questions remain to be resolved. For instance, how do the agents propagate information about the range of operations that they can perform? How does performance fall off as the nesting of queries, and the number of agents involved, increases? How do we prevent deadlock or circularity (ie a query being propagated back to its source)? Further developments in these areas will aid in the practical development of complex collaborative models and provide a theoretical foundation for some of the above conclusions.

For the moment we assume that the agents function within a very restricted context, but questions such as these are crucial if our approach is to extend into broader domains.
Eucalyptus regnans in Tasmania

Description
A very tall tree with rough fibrous bark to about halfway up trunk, then smooth, white or grey-green bark above, with umbels paired in leaf axils. The wood has been used for building, flooring, furniture, plywood and paper-making, and is moderately strong and hard though not durable. The tallest tree species in Australia, and the tallest hardwood in the world.

Species Distribution

Resources
- Australian Botanical Gardens
- Environment Australia

Report produced by Plant Species Report Generator

Figure 5: Final report of the distribution of Eucalyptus regnans in Tasmania. Generated by several distributed query agents.

Bibliography